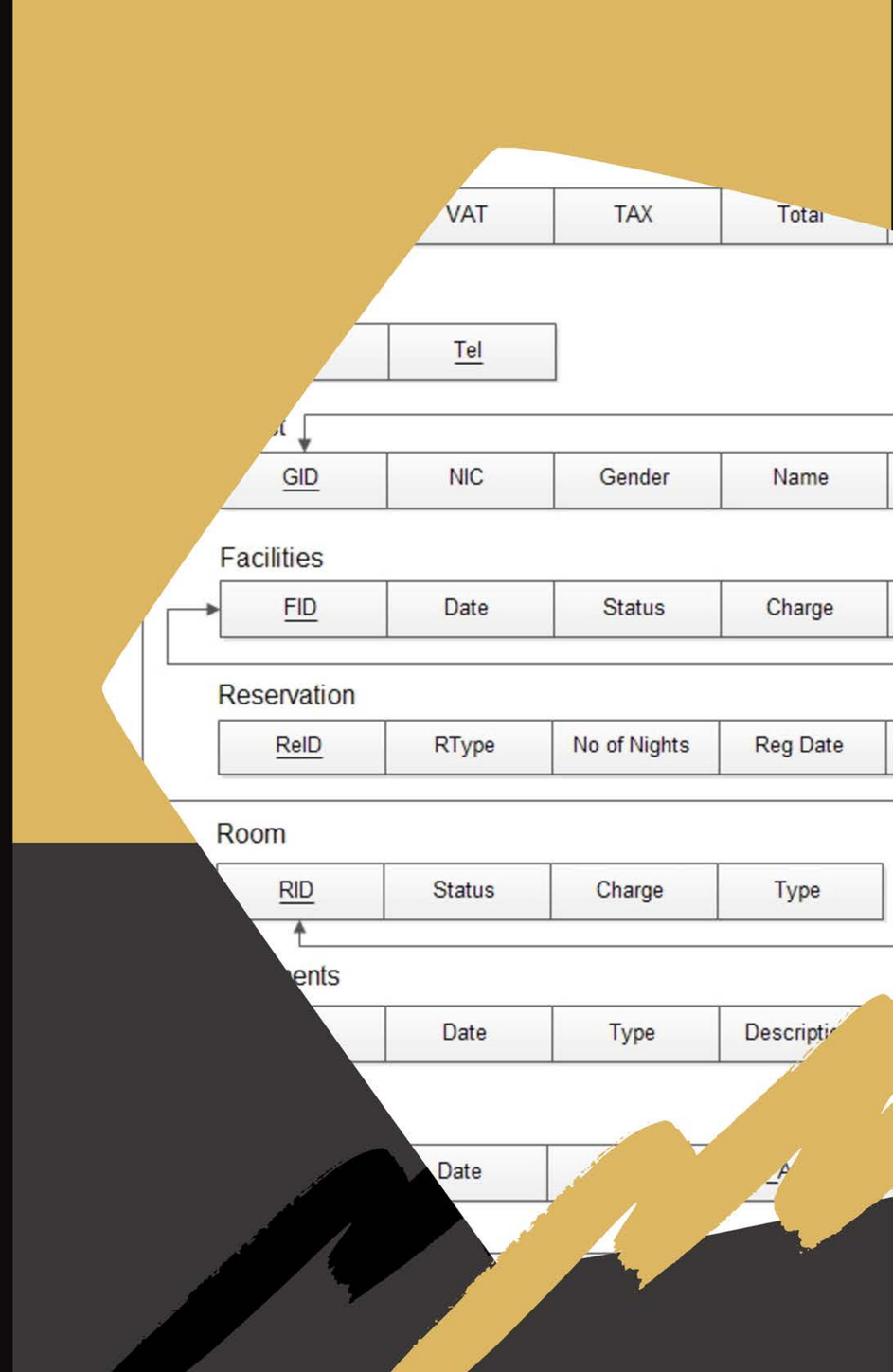
CSE370: Database Systems

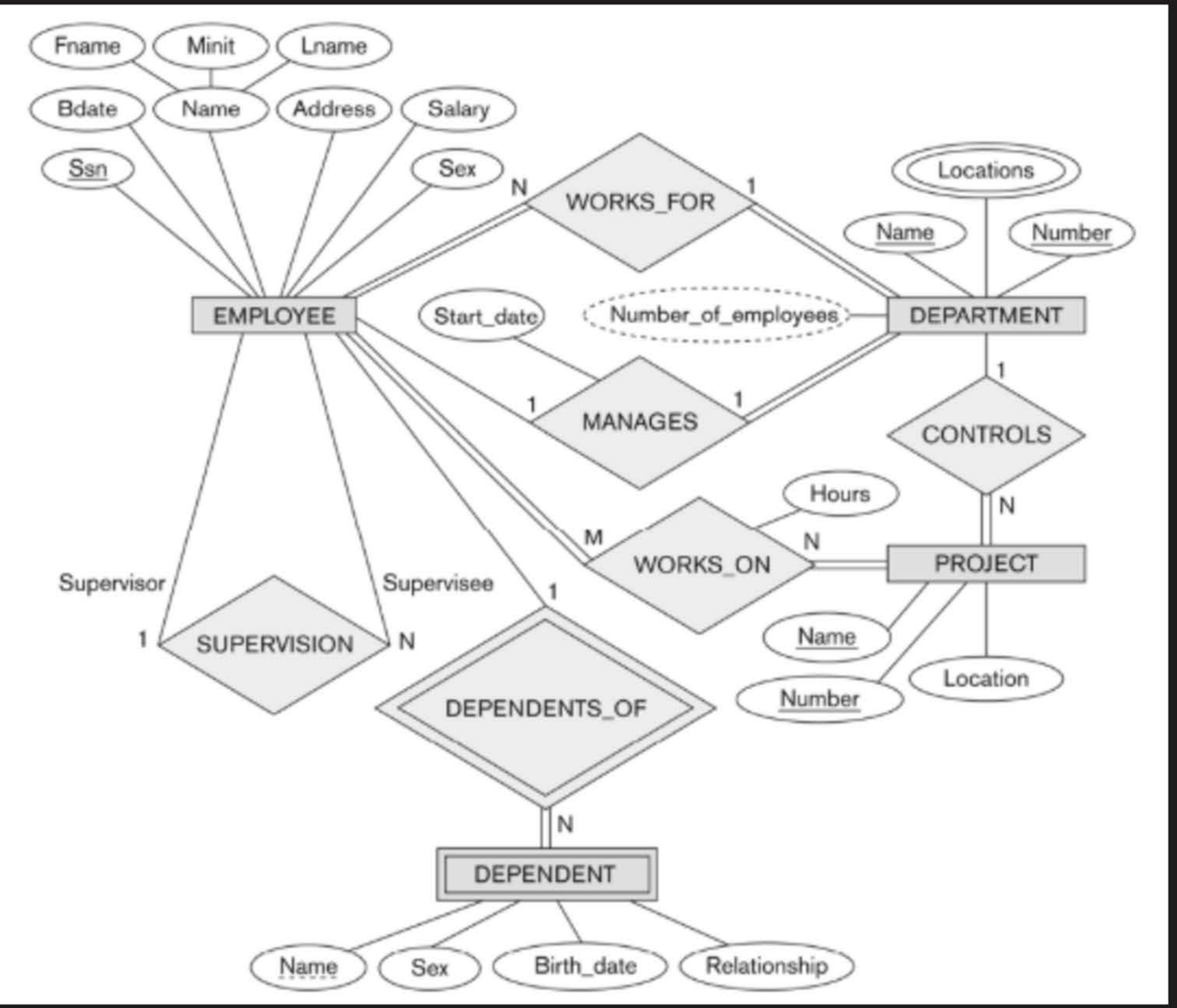
Lecture 5: Relational Database Design

Mapping ER/EER to Relational Schema



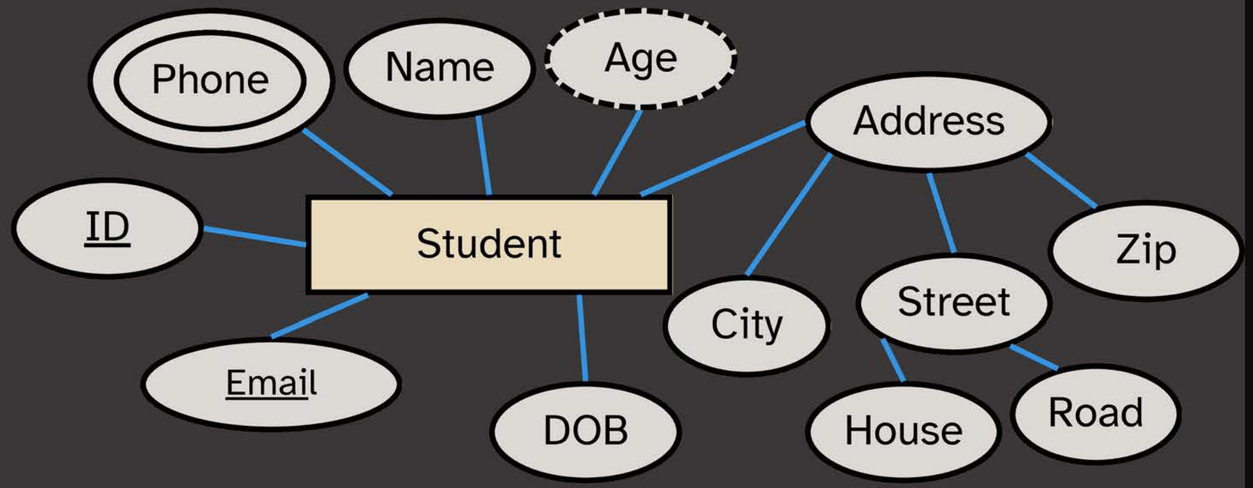
BRAC UNIVERSITY
Inspiring Excellence

ER to Relational Schema



- Step 1: Regular Entity Types
- Step 2: Weak Entity Types
- Step 3: Binary N:M Relationship Types
- Step 4: Binary 1:N Relationship Types.
- Step 5: Binary 1:1 Relationship Types.
- Step 6: Multivalued attributes.
- Step 7: N-ary Relationship Types.

Step 1: Mapping of Regular Entity Types

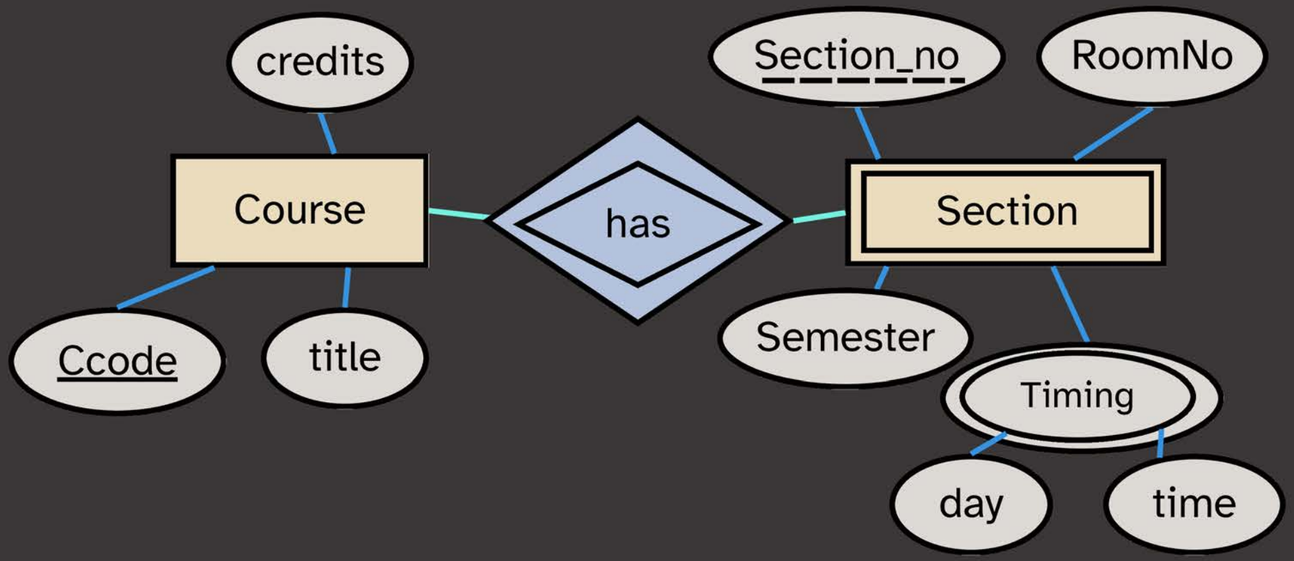


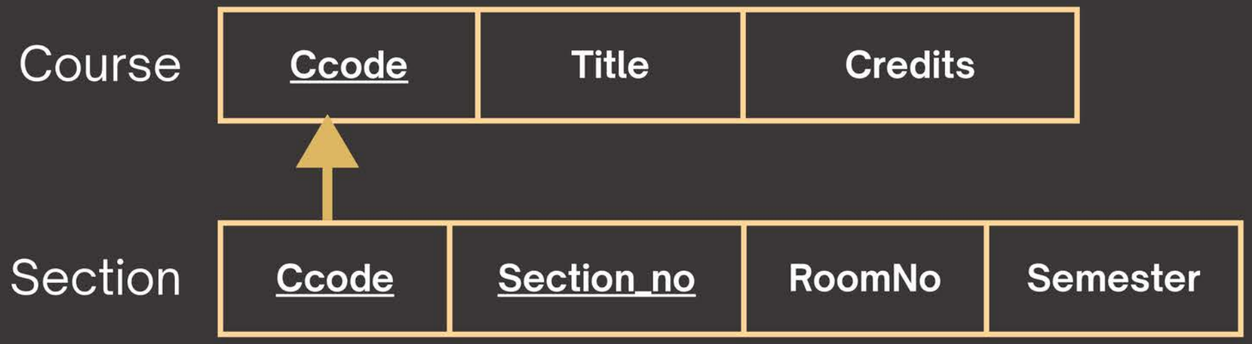
Student

ID	Name	Email	DOB	City	House	Road	Zip

- ✓ Create a separate table for each regular entity type. Tables must have names.
- ✓ Simple attributes will be added as columns. Simple components (e.g. city, house, road, zip) of composite attributes (like address) will be added as separate columns.
- ✓ Multivalued attributes like “phone” will be mapped later in step 6.
- ✓ Derived attributes like “Age” are not shown in the schema
- ✓ Only 1 candidate key will be selected as the primary key (pk) and it will be underlined in the schema. E.g. here Email is not selected as pk, so it is not underlined. If the pk is a composite attributes, then all simple components must be underlined.

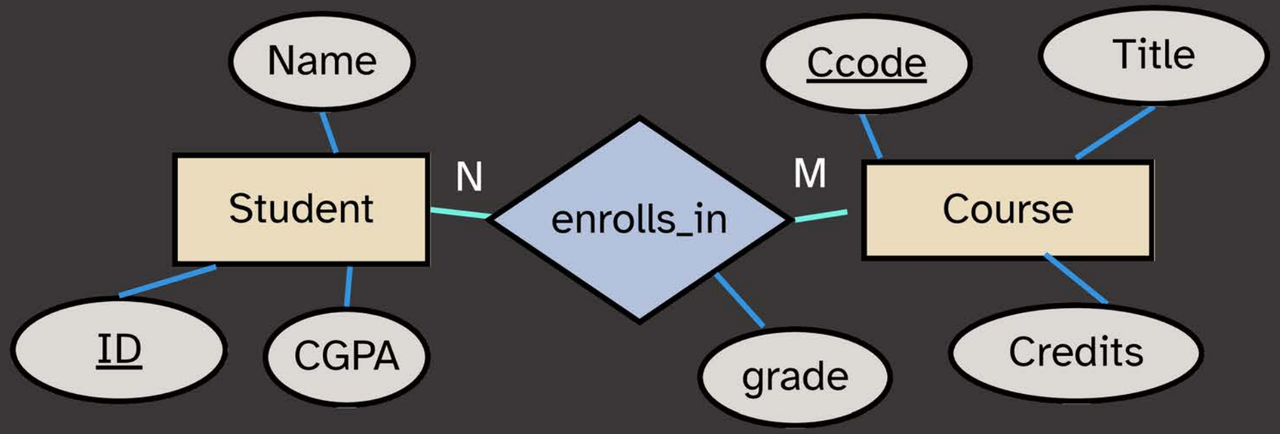
Step 2: Mapping of Weak Entity Types

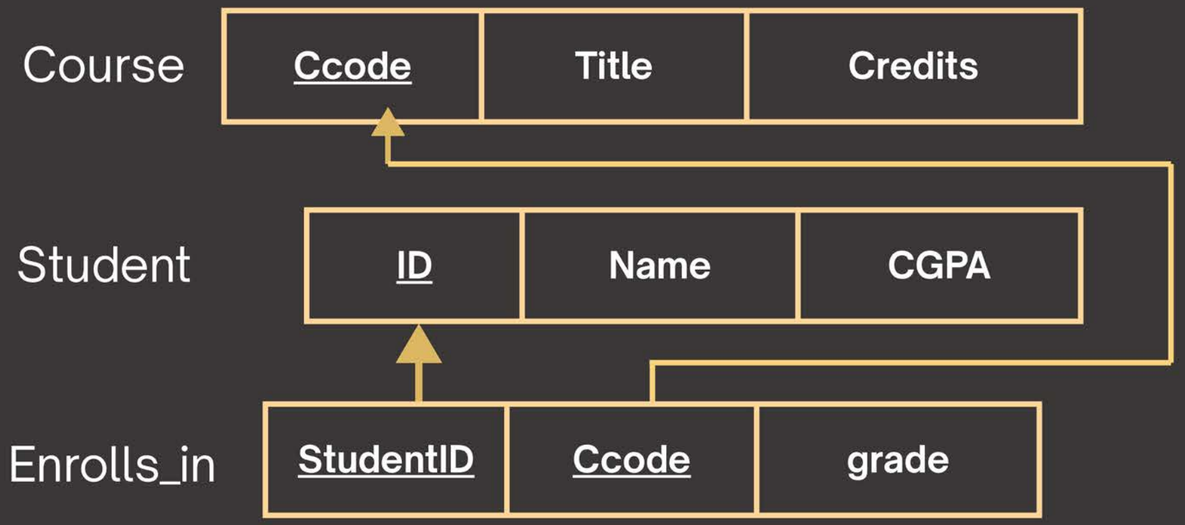




- Create a separate table for each weak entity type.
- Add any simple attributes and any simple components of composite attributes same as “Step 1”. Multivalued attributes (including multivalued composite attributes) will be mapped later.
- The weak entity must have an identifying relationship with its owner entity. Add the owner entity’s pk as a foreign key (fk) in the weak entity. FKs are shown using arrows. Here “Ccode“ from Course is added in Section as fk. If the pk was composed of several attributes, then all of them as a whole must be added as fk.
- The foreign key from the owner entity (e.g. Ccode) and the partial key of the weak entity (e.g. Section_no) will be the combined primary key of the weak entity and so both are underlined.

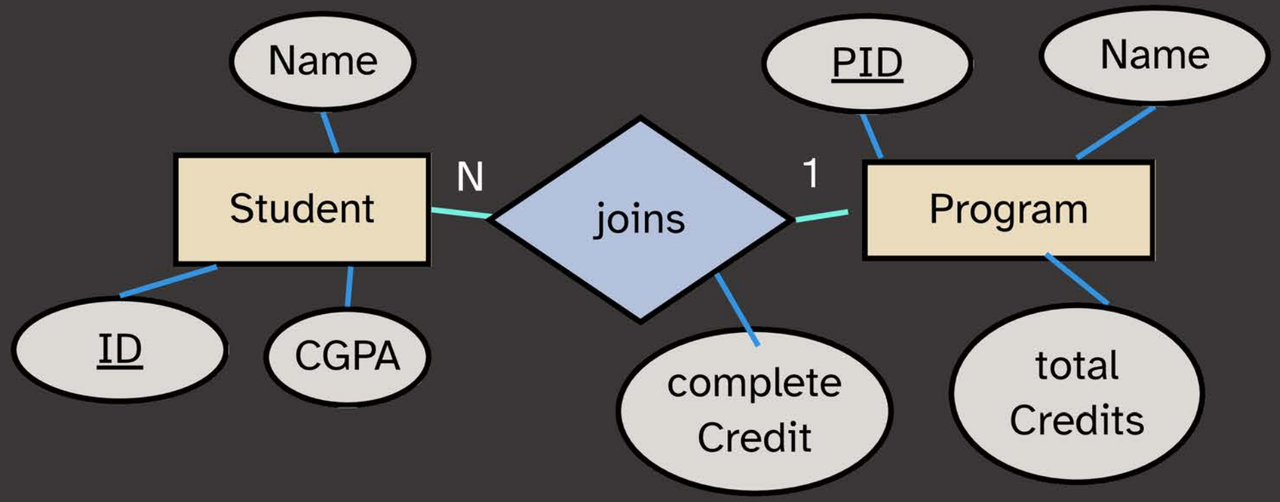
Step 3: Mapping of Binary N:M Relationship Types

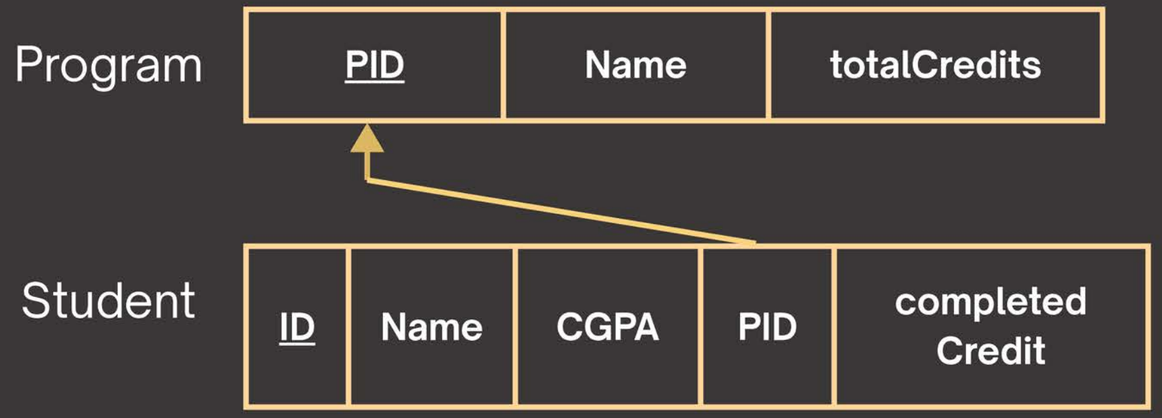




- ✓ Create a separate table for each N:M binary relationship (e.g. enrolls_in).
- ✓ The primary keys (pk) of both of the connected entity types (e.g. Course and Student) are added as foreign keys (fk) in the relationship table.
- ✓ The foreign keys are underlined in the relationship table as the combined primary key.
- ✓ Add any simple/composite relationship attributes in the table.
- ✓ This method of creating separate table for the relationship is called the cross-referencing approach

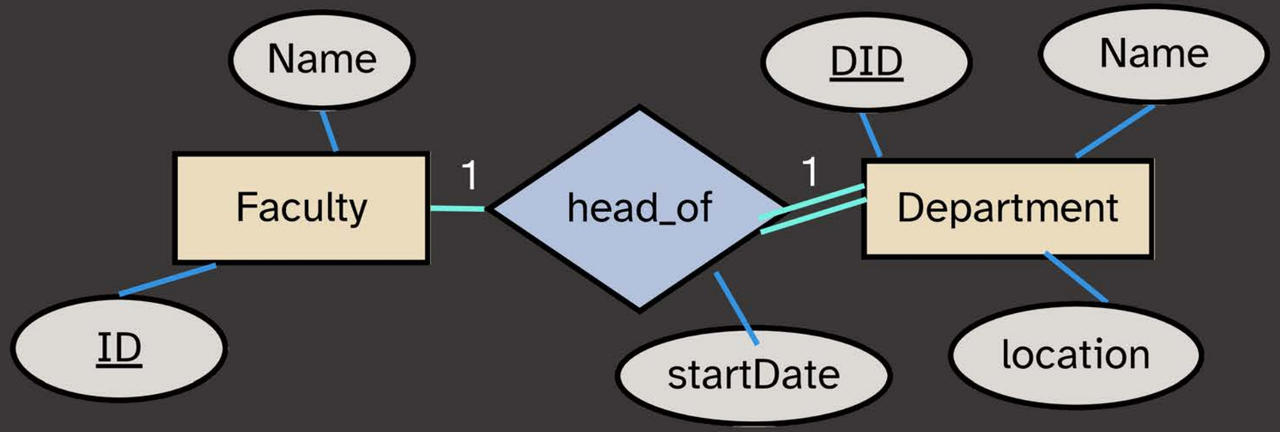
Step 4: Mapping of Binary 1:N Relationship Types

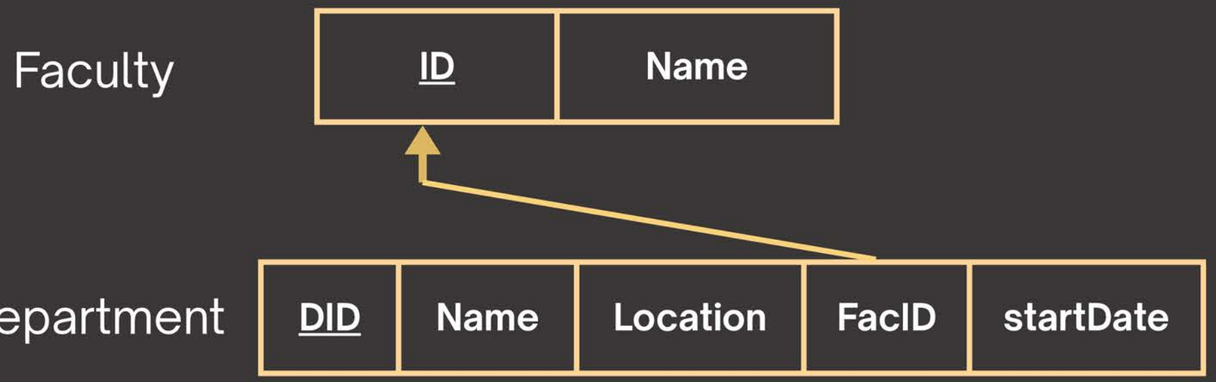




- ✓ 1:N relationships maybe mapped using cross-referencing approach. However, it is very inefficient, instead, the foreign key approach should be used which is space efficient.
- ✓ For every 1:N relationship (e.g. joins), take the primary key of entity table connected to the “1” side (e.g. Program) and add it as the foreign key to the entity table connected to the “N” side (e.g. Student) of the relationship.
- ✓ All relationship attributes (except multivalued attributes) are also added to the entity table connected to the “N” side of the relationship.

Step 5: Mapping of Binary 1:1 Relationship Types

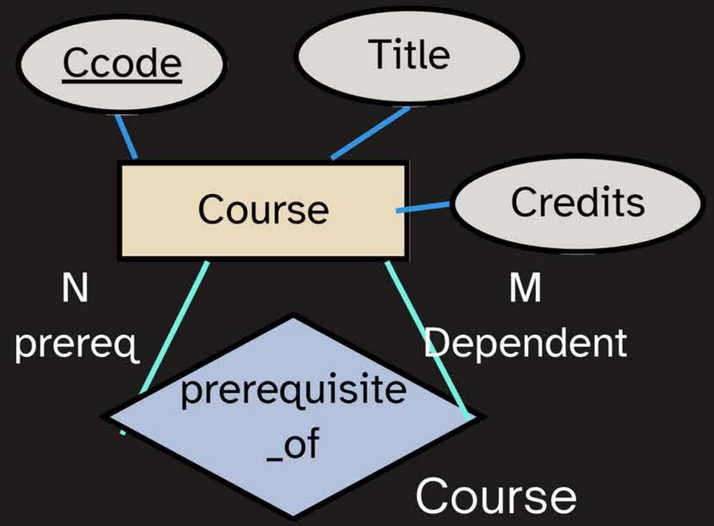




- There are 3 possible approaches for mapping 1:1 binary relationships: cross-referencing approach, merged relation option, foreign key approach.
- The cross-referencing approach will follow the exact same process as Step 3.
- In merged relation option the two entity type tables are merged together to form a single table and the primary key is the combination of the two key attributes. This option may lead to storage wastage and is not the most efficient option.
- For the foreign key approach (shown on left) take any one of the entity table's primary key and add it as foreign key to the other entity table. The relationship attributes should be added to the same table as the foreign key. It is most efficient to add the foreign key on the entity table that has "total" participation as it will prevent unnecessary null values.

How To Map Recursive Relationships ?

Recursive relationships will be mapped same way as binary relationships- N:M → cross-referencing method (shown left), 1:N → foreign key approach (shown right), 1:1 → cross-referencing or foreign key approach

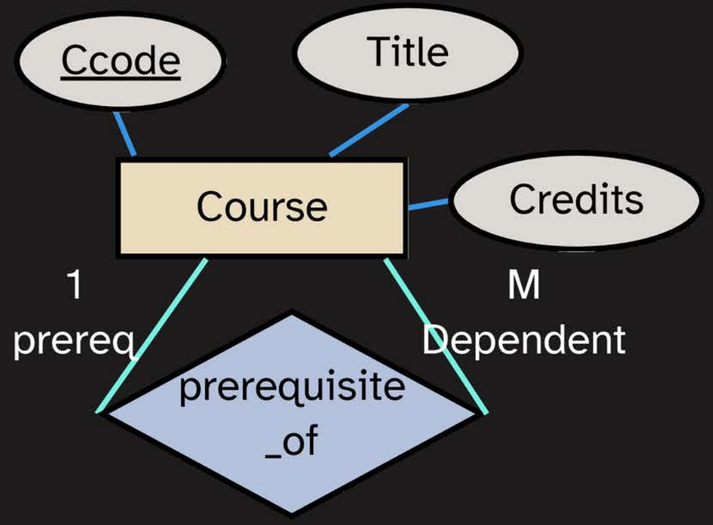


Course

Ccode	Title	Credits

Prerequisite_of

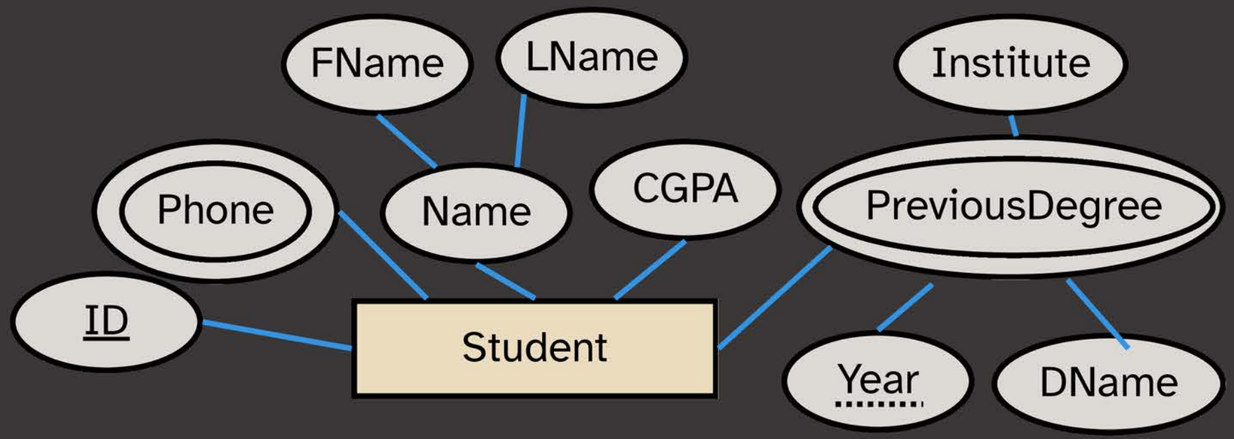
prereq_Ccode	dependent_Ccode



Course

Ccode	Title	Credits	prereq_Ccode

Step 6: Mapping of Multivalued Attributes



Student

ID	FName	LName	CGPA

Student_Phone

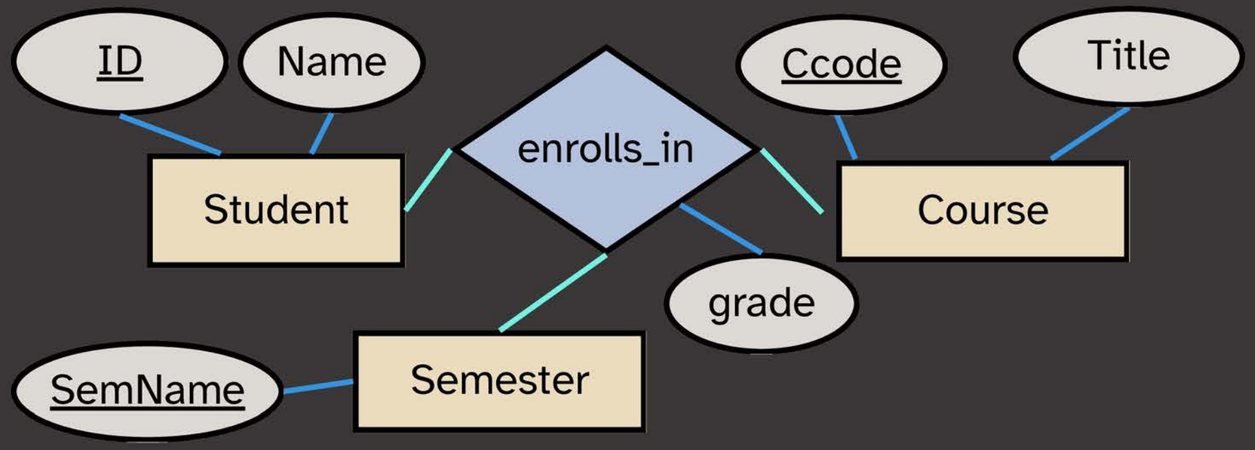
ID	Phone

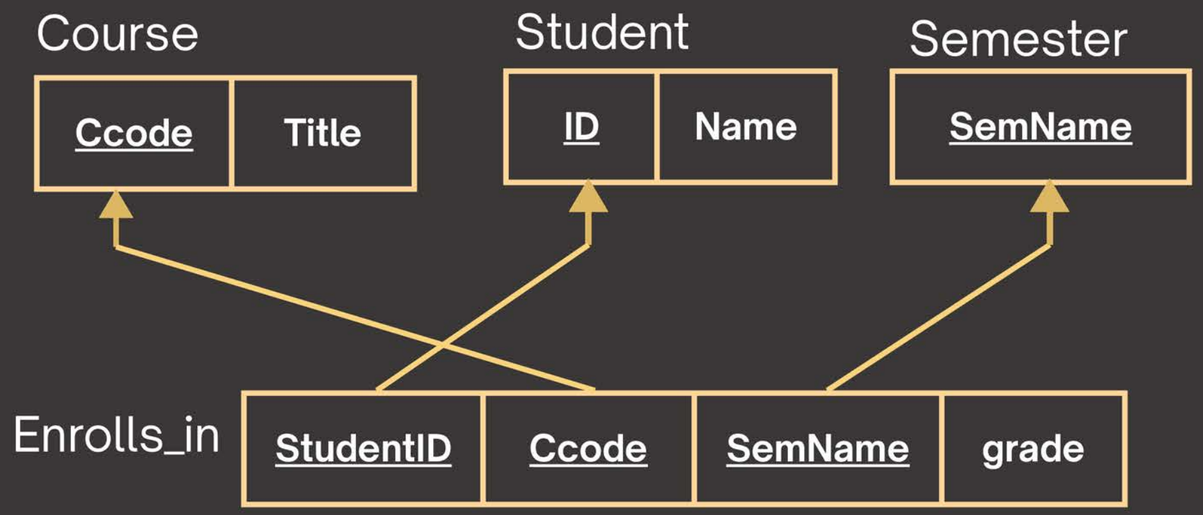
PreviousDegree

ID	Year	DName	Institute

- ✓ Create separate tables for each multivalued attributed (including multivalued-composite attribute (e.g. for Phone and PreviousDegree).
- ✓ Add the primary key of the entity table that the attribute belongs to (e.g ID from Student) as the foreign key in the multivalued table.
- ✓ If it is only multivalued (like Phone), then add another column for storing the value of that attribute. Underline the foreign key and the attribute column as the combined primary key (e.g. both ID and Phone are underlined in Student_Phone table).
- ✓ If it is a multivalued-composite attribute (like PreviousDegree), add only the simple components of the attribute as separate columns. Underline the foreign key and any component indicated as partial key as the combined primary key. Note: If partial key component is not shown, then underline all components as part of the primary key.

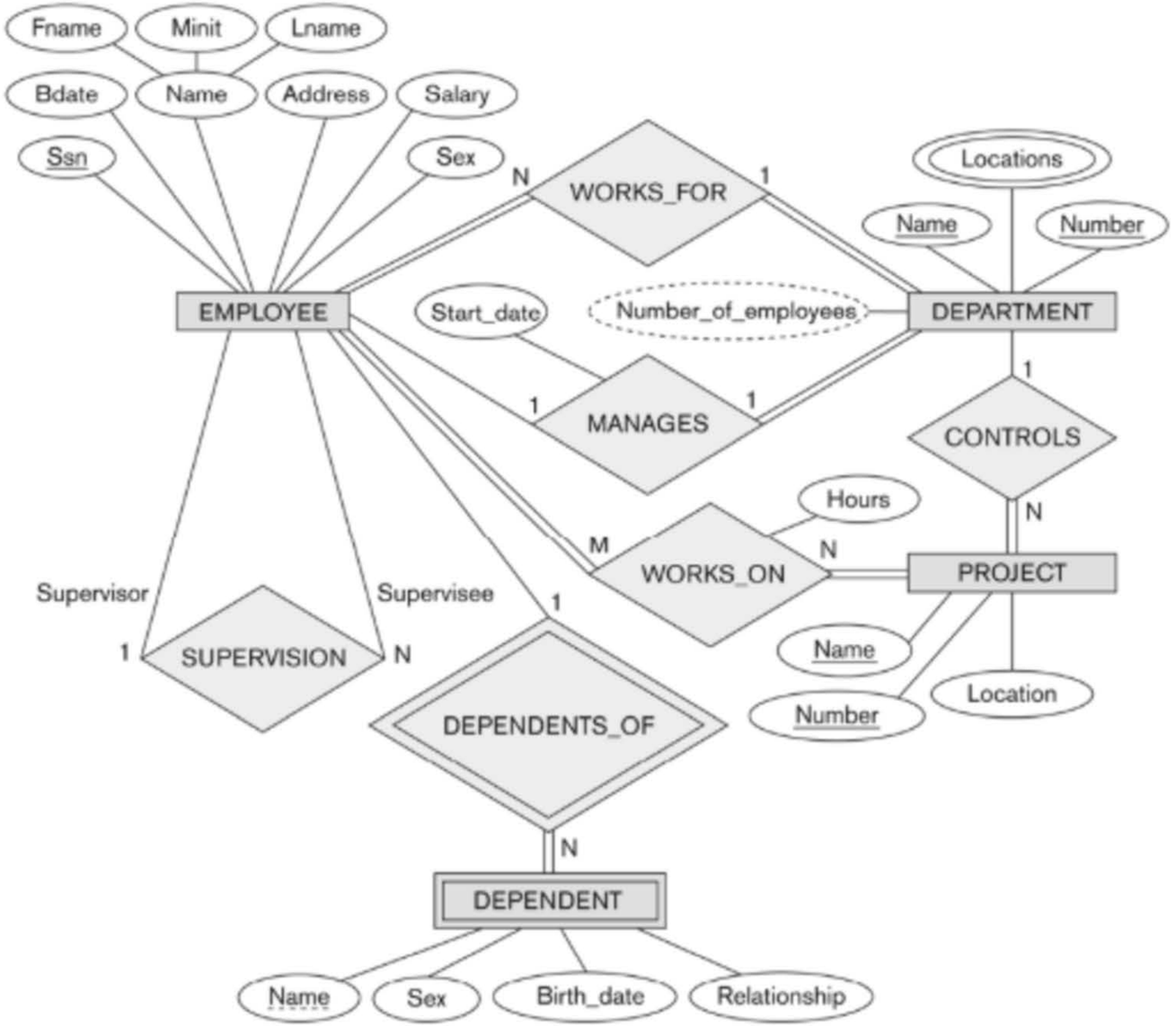
Step 7: Mapping of N-ary Relationship Types



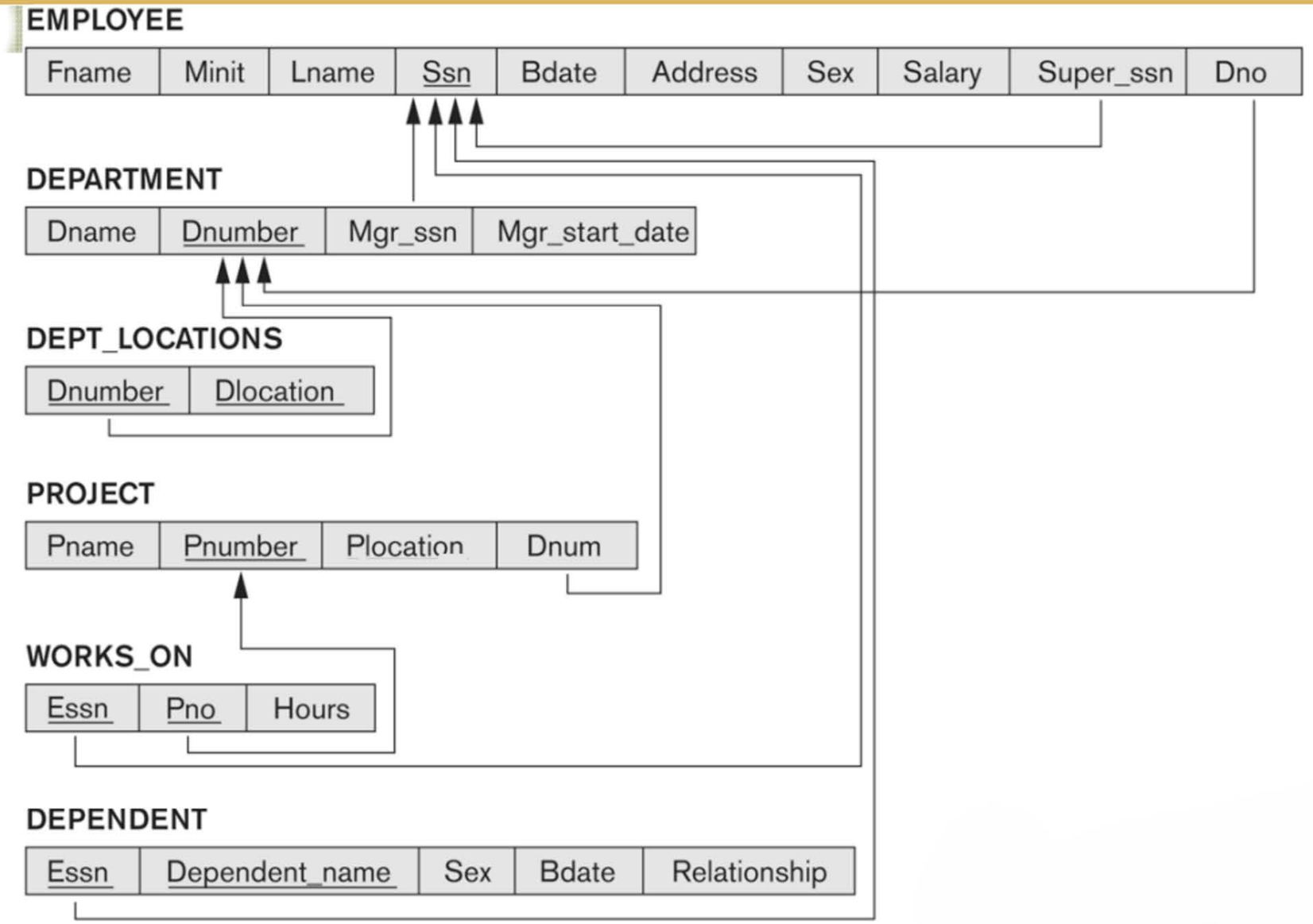


- ✓ Create a separate table for each N-ary (or ternary) relationship.
- ✓ The primary keys (pk) of all of the connected entity types (e.g. Course, Student, Semester) are added as foreign keys (fk) in the relationship table.
- ✓ The foreign keys are underlined in the relationship table as the combined primary key.
- ✓ Add any simple/composite relationship attributes in the table.

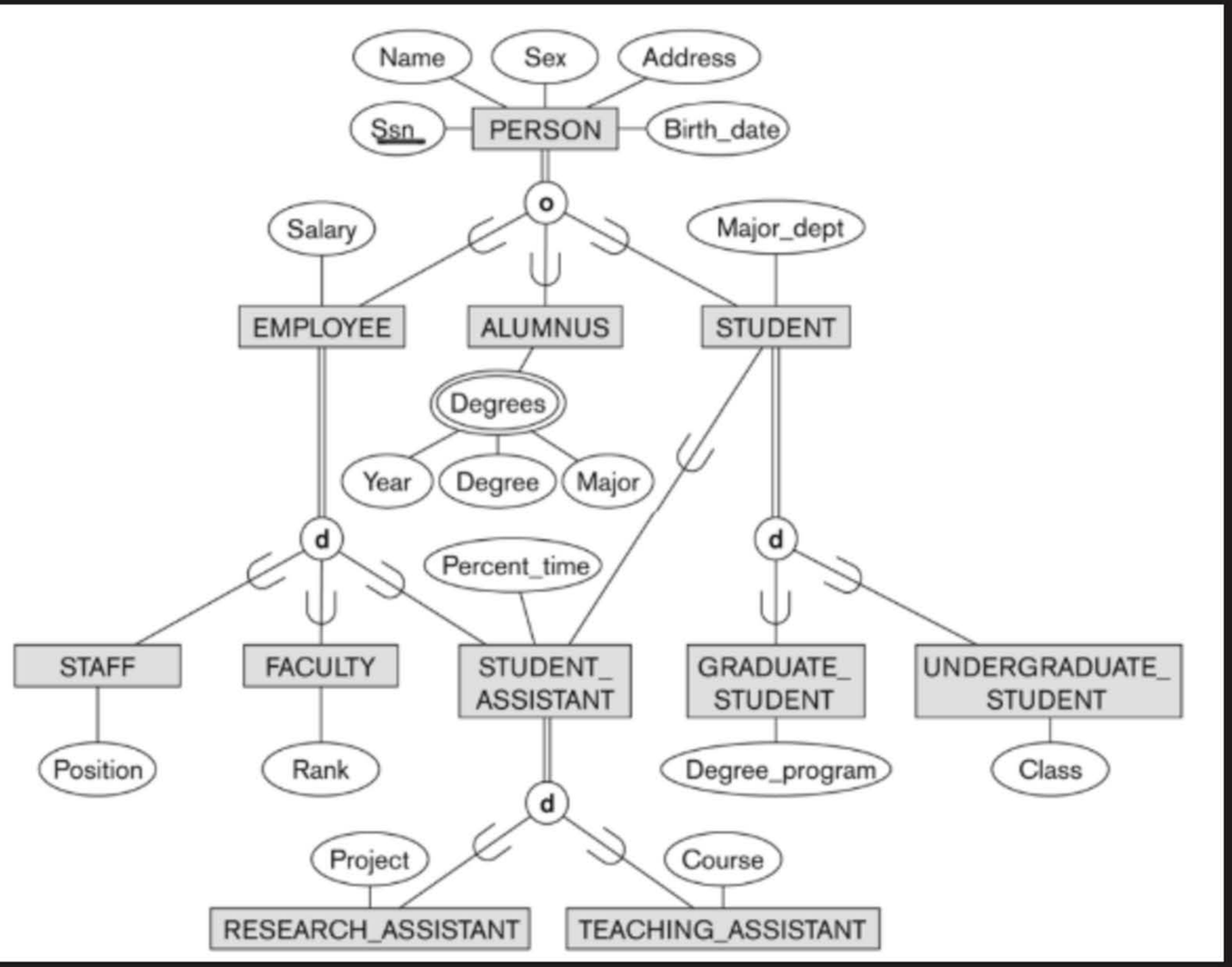
ER to Relational Schema in Practice (1)



ER to Relational Schema in Practice (2)

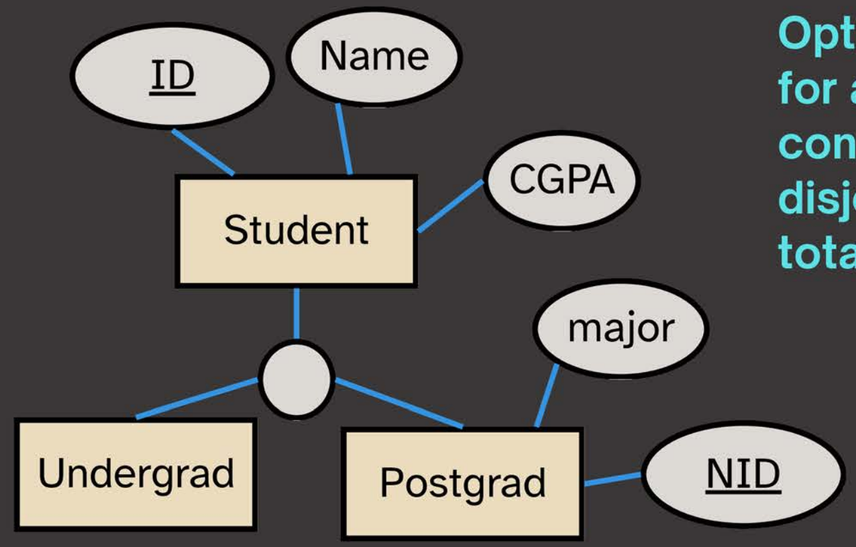


EER to Relational Schema

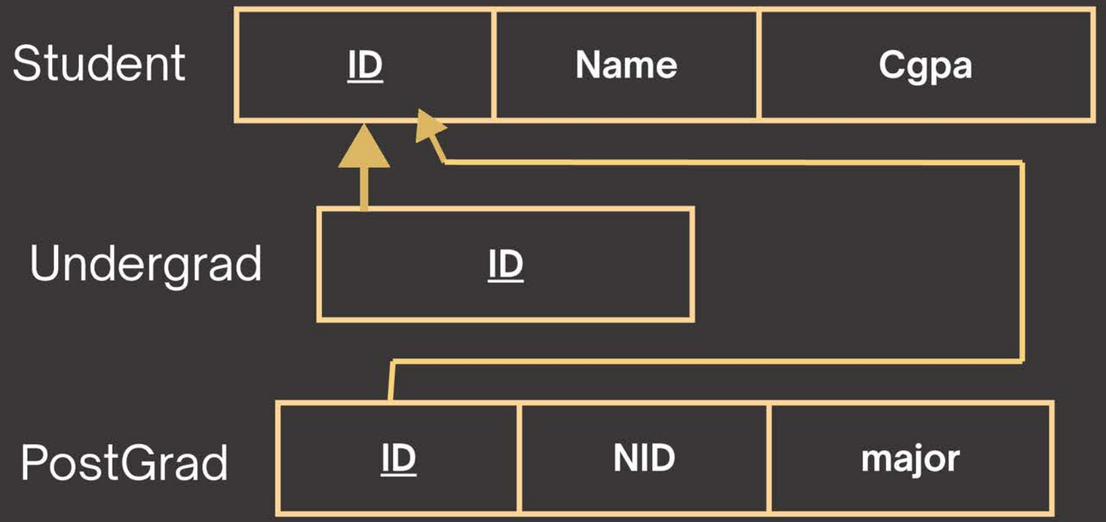


- Step 8: Specialization/Generalization
- ✓ 8A: Multiple tables - superclass & subclass
- ✓ 8B: Multiple tables - subclasses only
- ✓ 8C: Single table with 1 type attribute
- ✓ 8D: Single table with many type attributes

Step 8: Mapping of Specialization or Generalization (8A)



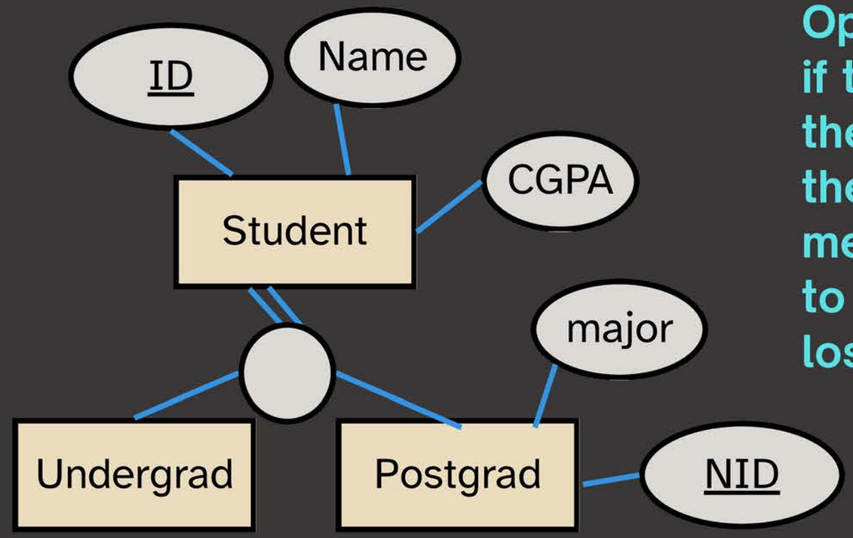
Option 8A, is applicable for all specialization constraints: disjoint/overlapping and total/partial.



Option 8A: Separate tables for the superclass and each of the subclasses.

- ✓ Superclass table is created exactly like step 1. The primary key of the superclass table is added as foreign keys in the subclass tables.
- ✓ The foreign keys from the superclass also become the primary keys of the subclasses.
- ✓ The local attributes from the subclasses are added to their respective tables.

Step 8: Mapping of Specialization or Generalization (8B)



Option 8B, is only applicable if the specialization is total. If the specialization is partial, then data of superclass members that do not belong to any subclasses will be lost.

Undergrad

ID	Name	CGPA

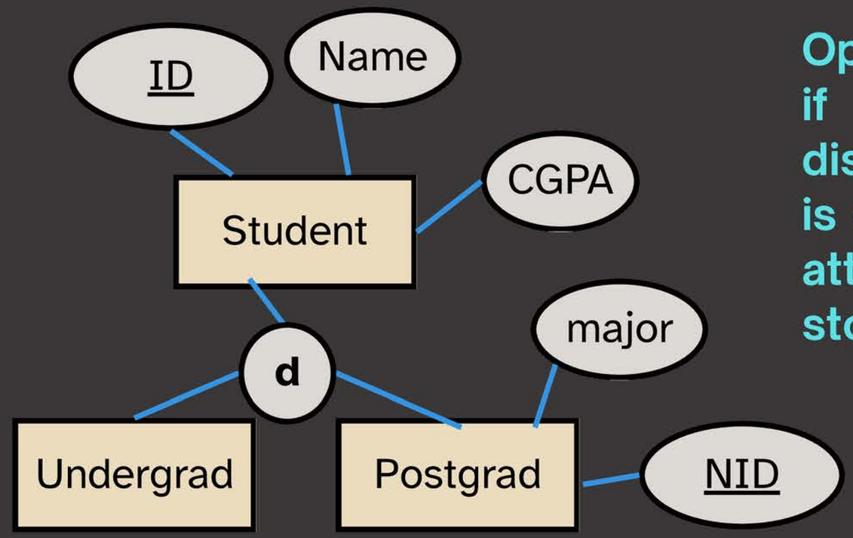
PostGrad

ID	Name	CGPA	NID	major

Option 8B: Tables only for each subclass. No table for superclass.

- ✓ All superclass attributes are added to each subclass table. One key attribute from the superclass is selected as primary key in each subclass.
- ✓ The local attributes from the subclasses are added to their respective tables only.
- ✓ If superclass has any multivalued attribute, then step 6 will be followed for each subclass. For example, if Student has a multivalued phone attribute, then Phone table will be created twice, once for undergrad and once for postgrad

Step 8: Mapping of Specialization or Generalization (8C)



Option 8C, is only applicable if the specialization is disjoint. If the specialization is overlapping, then one type attribute is not sufficient to store several subclass info.

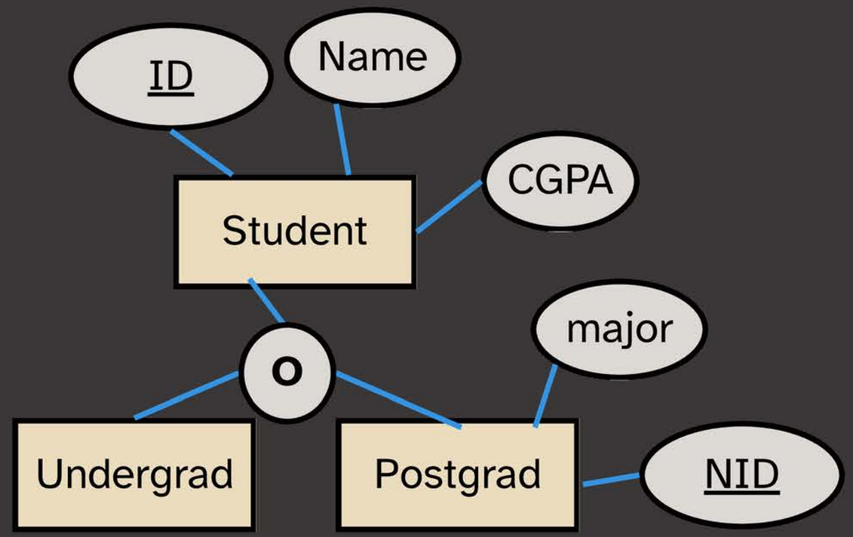
Student

ID	Name	CGPA	NID	major	StudentType

Option 8C: Single table for superclass and subclass with one type attribute.

- ✓ One table is created for superclass and subclasses.
- ✓ Attributes of superclass and all subclasses are added to the same table. One key attribute from superclass is selected as primary key.
- ✓ An additional “type” attribute is added to the table (e.g. StudentType). This attribute is used as a discriminator to indicate which subclass the entity belongs to.

Step 8: Mapping of Specialization or Generalization (8D)



Option 8D, is applicable for all specialization constraints: disjoint/overlapping and total/partial. However, it is inefficient for disjoint specializations, because one additional column is sufficient for disjoint, several additional columns will waste space.

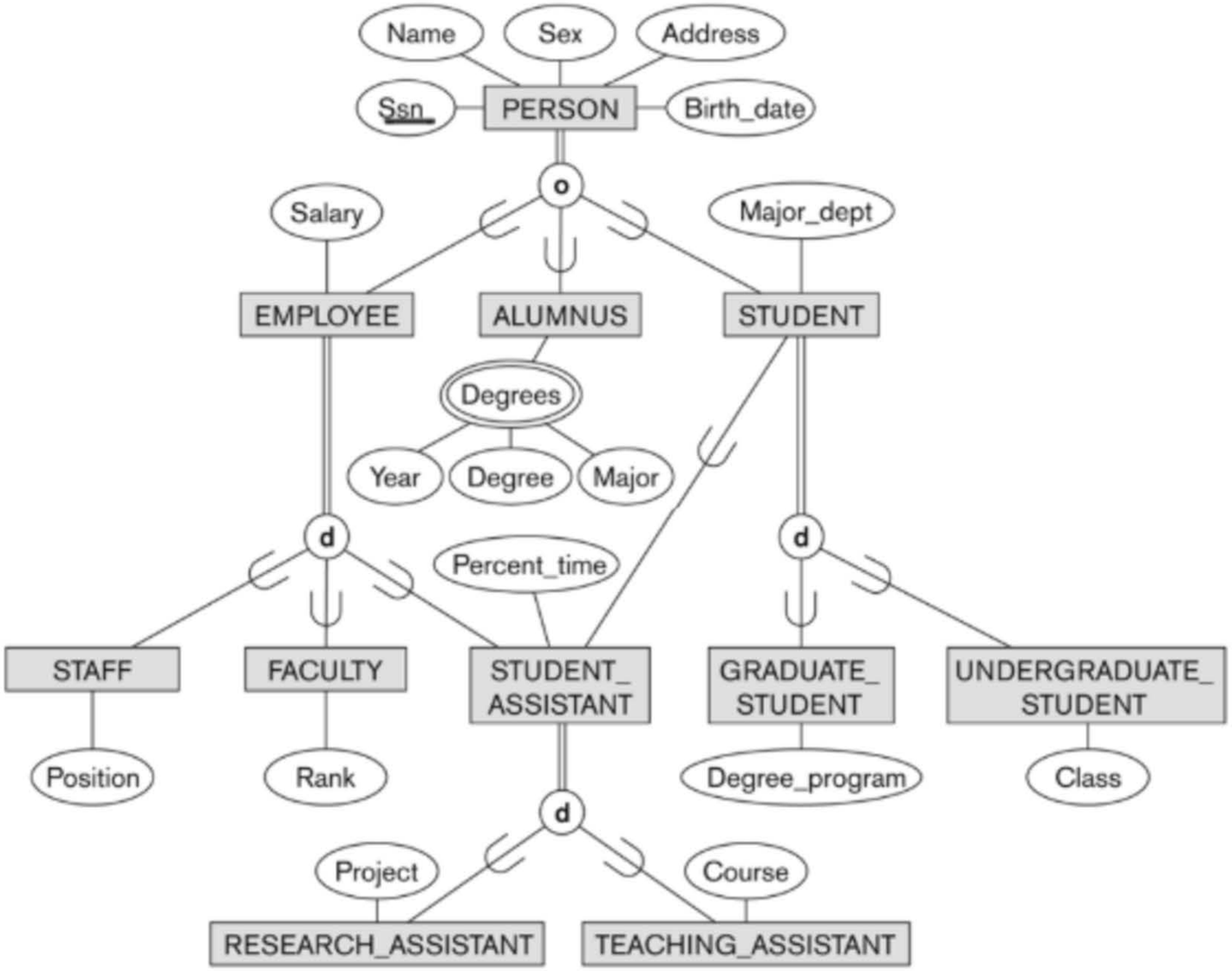
Student

ID	Name	CGPA	NID	major	UGFlag	PGFlag

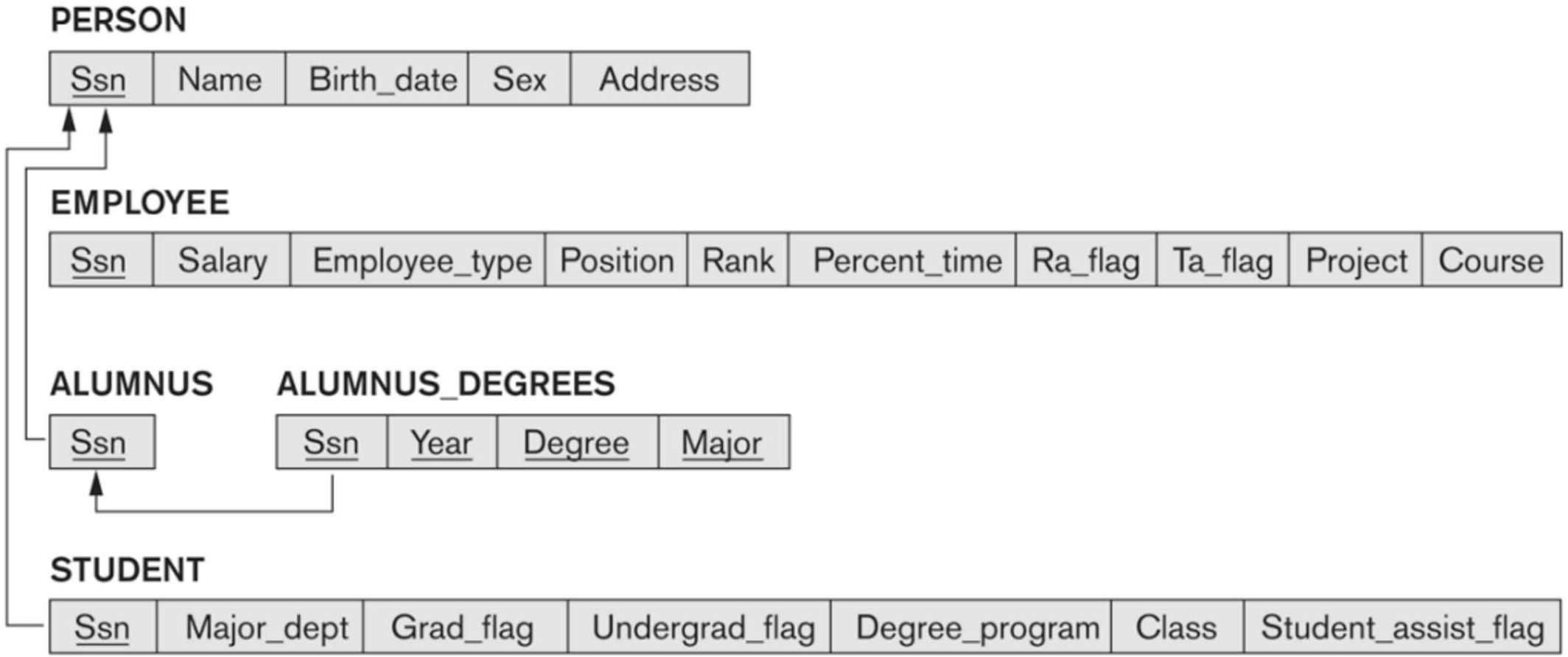
Option 8D: Single table for superclass and subclass with multiple type attribute.

- ✓ One table is created for superclass and subclasses.
- ✓ Attributes of superclass and all subclasses are added to the same table. One key attribute from superclass is selected as primary key.
- ✓ “N” additional type or “flag” attributes are added to the table (e.g. UGFlag, PGFlag), where N is the number of subclasses in that specialization. These attributes store boolean values (true/false) to indicate if an entity belongs to a subclass or not.

EER to Relational Schema in Practice (1)



EER to Relational Schema in Practice (2)



What Next?

LECTURE 6: NORMALIZATION

LOADING.....